Part 3: Radial Basis Functions

December 28, 2020

I would like to quickly thank everyone for reading my last post about Gaussian processes and Bayesian optimization. It was a lot of fun to do and I hope everyone likes the code I provided. This week we are going over radial basis functions (RBFs). An RBF is a model structure that, much like Gaussian processes, relates some input of interest x to some output of interest y by relating a value of interest x to training data X and Y. In this case, that "relationship" is radial, in that it is distance based, such as with the much-beloved L2 norm $r = ||x - X||_2^2$. Point that are closer have smaller r values, so are going to be modeled as more similar to one another. This is quite similar to Gaussian processes, so hopefully the qualitative concept is fresh in everyone's mind.

1 Radial Basis Function Model

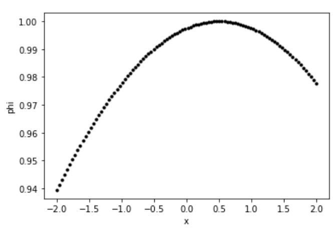
The basic model of an RBF is as follows:

$$\hat{y} = \Sigma_{i=1}^{n_c} \lambda_i \phi(r_i) \tag{1}$$

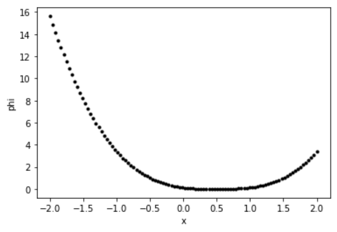
$$r = ||\hat{x} - x_c||_2^2$$

Notice we have added ϕ, which is the activation function that takes some r and projects it into output space. Common activation functions are exponential/Gaussian $\phi(r) = exp(-\sigma^2 r)$, or the cubic $\phi(r) = r^3$, or the thin plate spline $\phi(r) = r^2 ln(r)$. In Figures 1 and 2 we can see that as one moves away from the "center" located at $x = 0.5$, or as r increases, ϕ changes as well.

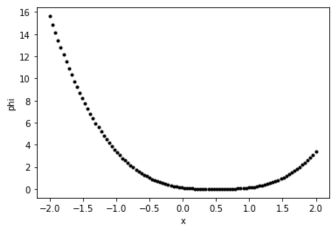
As a note, it doesn't matter if ϕ increases or decreases as a function of r because in Equation (1), when we chain a bunch of these ϕ's together, they get multiplied by coefficients λ which can take on any real value to account for the activation function being an arbitrary sign.



(a) Gaussian

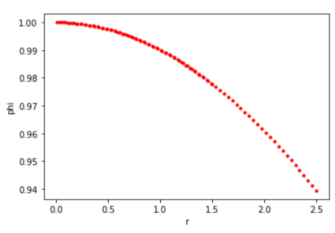


(b) Cubic

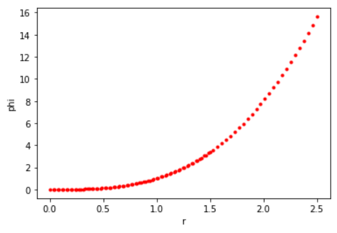


(c) Thin Plate Spline

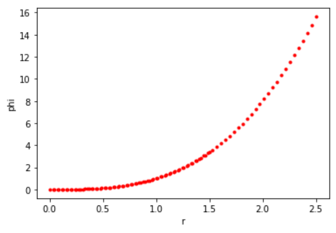
Figure 1: RBFs in x space with center at $x = 0.5$



(a) Gaussian



(b) Cubic



(c) Thin Plate Spline

Figure 2: RBFs in r space

2 Toy Problem

Now that we have seen the RBF work on a single point, let's take a toy example problem $f(x) = x^2 + x + 1 + U(0, y/2)$ and model it using the gaussian RBF. This can of course be generalized to multi-dimensional problems by making r reflect a multi-dimensional distance metric, but for simplicity (and Christmas Spirit^{TM}) we will be in a single dimension.

Using Equation (1) as a guide, we know that to solve linear regression problems that "look" like an RBF activation chain we solve $w = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T Y$. Therefore, defining $\boldsymbol{\Phi}(X) = \phi(x, x')$ for all $x, x' = 1...N$ training data, we can solve for the coefficients in Equation (1). In the same way we solved for the maximum likelihood of a multi-dimensional linear regression problem, we solve for the maximum likelihood of a multi-point transformed regression problem.

$$\lambda = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T Y \tag{2}$$

You will sometimes see Equation (1) written in matrix form with $\boldsymbol{\Phi} = \phi(x_i, X_j)$ for testing data x_i and training data X_j. I will drop the bolding now, I just wanted to make it clear we are talking about matrices. Below I have provides some examples of 3 training data points on the toy problem where the solution λ has been solved for and predictions $\hat{y}$ are in red.

$$\hat{y} = \Phi\lambda$$

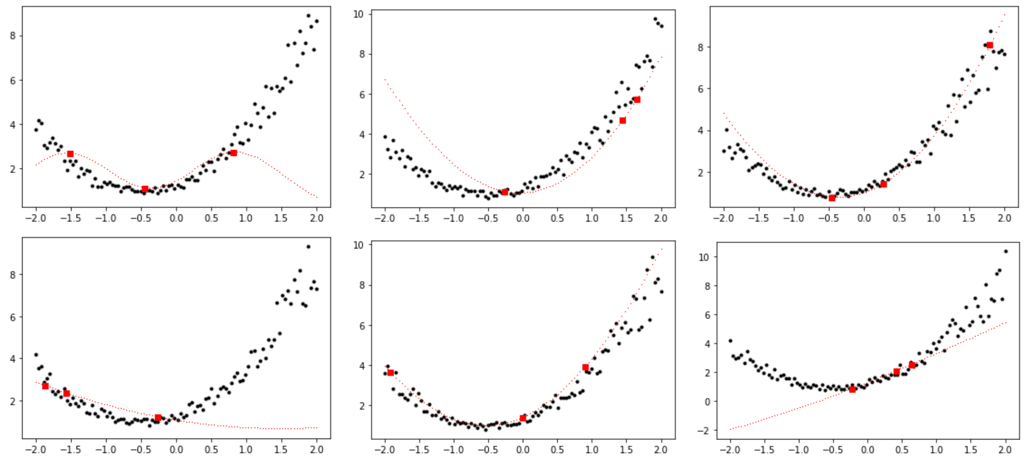


Figure 3: Regression w/ Random Training Data

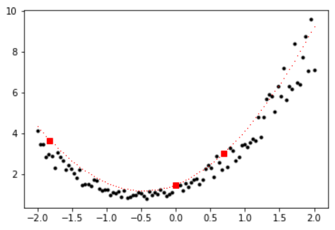
3 Classification

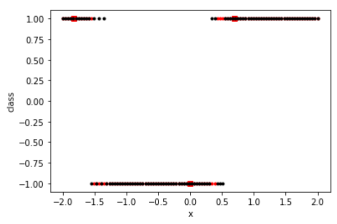
If we would like to solve a classification problem, say $y \in \{1, -1\}$, we modify Equation (1):

$$\hat{y} = sign\{\Sigma_{i=1}^{n_c} \lambda_i \phi(r_i)\} \tag{3}$$

$$y(x) = 1 * \{x^2 + x + 1 \geq 2\} \tag{4}$$

In this example, the actual function $f(x)$ is hidden and we only have access to the class $y \in \{1, -1\}$. Here we train λ and predict $\hat{y}$ the same as in regression problems. In Figure 4 a regression problem is posed as the classification problem.


(a) Regression



(b) Classification

Figure 4: Classification w / Random Training Data | red squares = training data, red dots = predictions

4 Polynomial Terms

A common practice to improve prediction with RBFs is to add a polynomial term:

$$\hat{y} = \Sigma_{i=1}^{n_c} \lambda_i \phi(r_i) + p(x) \tag{5}$$

So for example we would have $\hat{y} = \Sigma_{i=1}^{n_c} \lambda_i \phi(r_i) + c_0 + c_1(x)$ for a linear term and bias for a single variable. To solve for this we solve $A\Lambda = \gamma$ with P being rows of training data $[1 \; X_1 ... X_N]$ and $\gamma = [y \; 0]$. The regression results are shown.

$$A = \begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix}$$

$$\Lambda = \begin{pmatrix} \lambda \\ c \end{pmatrix}$$

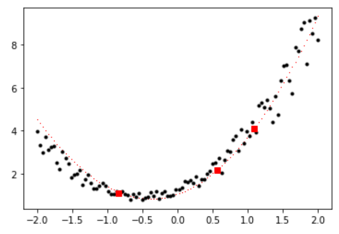


Figure 5: Regression w / RBF + $c_0 + c_1 x$ Term

5 Hyperparameters for RBFs

We will then chose the best hyperparameters σ by varying σ and picking the parameter with the lowest out-of-sample error. Here I will skip any cross-validation procedure and show the correct randomized error for various values.

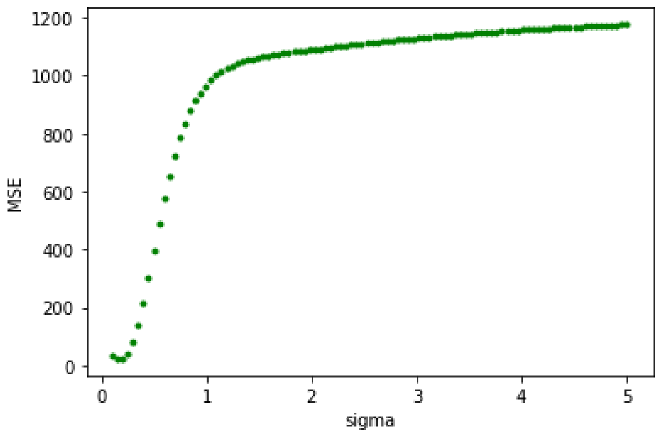
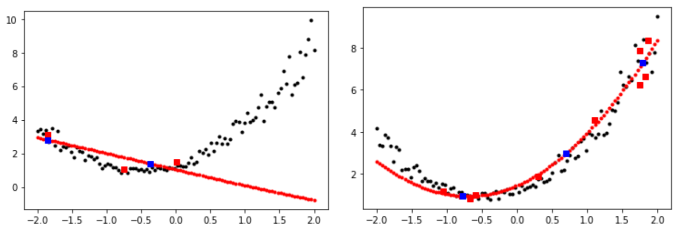


Figure 6: Mean Squared Error for Various σ in RBF Model

We should also look at the number of centers. So far the centers for the RBF have all been the training data, but these can be placed anywhere. Generally, the more centers, the more fit the data is. For example note Figure 7(a). With 2 centers, which is less than the 3 data points, the model becomes a simple linear model. We therefore seek to reduce the number of nodes to the minimum number required. Against one should use cross-validation techniques or other model comparison tools.



(a) 2 Centers (b) 3 Centers

Figure 7: RBF Regression for Various Numbers of Centers | data points = red, centers = blue

Okay thank you everyone for reading. Next week we will be talking about MCMC and Bayesian Methods a bit. Have a safe and fun holiday!